

### Remarks

Reconsideration of the application is requested.

Claims 1-23, 25-65, and 67-84 have been rejected. Claims 1, 10, 12, 18-19, 43, 52, 54, and 60-61 have been amended. Claims 3, 11, 45, and 53 have been cancelled. Accordingly, Claims 1-2, 4-10, 12-23, 25-44, 46-52, 54-65, and 67-84 remain pending in the application.

#### Claim Rejections – 35 U.S.C. § 112

In “Claim Rejections – 35 USC § 112,” item 5 on page 2 of the above-identified Office Action, claims 12-19 and 54-61 have been rejected as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention under 35 U.S.C. § 112, second paragraph. More specifically, the Examiner objects to the use of the terms “previously generated”, as in “previously generated proxy object.”

In response, Applicants have amended the above claims to delete the objected-to terms, thus obviating this rejection.

#### Claim Rejections – 35 U.S.C. § 103

To establish obviousness under 35 U.S.C. § 103, the Examiner must view the invention as a whole. Further, the Examiner is to perform the obviousness analysis in accordance with the standard set forth by the Supreme Court in *Graham v. John Deere Co.* That standard requires that the Examiner (1) determine the scope and content of the prior art; (2) ascertain the differences between the prior art and the claims in issue; (3) resolve the level of ordinary skill in the art; and (4) evaluate evidence of secondary considerations. 383 U.S. 1, 17-18 (1966); see also MPEP 2141. Secondary considerations include whether the invention met with commercial success, whether the invention answered a long felt need, and whether others attempting the invention have failed. *Graham*, 383 U.S. at 17-18. Further, in applying the *Graham* framework, the

Examiner must consider the invention as a whole, without the benefit of hindsight.  
MPEP 2141.

1. In “Claim Rejections – 35 USC § 103,” item 7 on page 3 of the above-identified Office Action, claims 1-3, 10-11, 43-45, and 52-53 have been rejected as being unpatentable over U.S. Patent No. 6,209,018 to *Ben-Shachar* (hereinafter “Ben-Shachar”) in view of “Applying C++, Patterns, and Components to Develop an IDL Compiler for CORBA AMI Callbacks,” *Arulanthu et al.* (hereinafter “Arulanthu”) under 35 U.S.C. § 103(a).

The rejections of claims 3, 11, 45, and 53 are obviated by their cancellations.

Claim 1 recites a “method of specifying an asynchronous web service within a procedural programming environment, the method comprising:

providing a source code representation of at least a portion of web service logic,  
the logic including at least one method declared to be a callback method;  
identifying a member variable declared to implement said callback method to  
cause a compiler to generate a client proxy object for interacting  
asynchronously with the client using said callback method, and to assign  
the client proxy object to said member variable; and  
specifying one or more declarative annotations associated with said callback  
method to cause a compiler to generate one or more persistent  
components to maintain conversational state related to the identified  
member variable.”

Thus, when amended claim 1 is viewed as a whole, as required by law, it is directed towards a novel method of specifying an asynchronous web service in a procedural programming environment. In addition to identifying a member variable to implement a callback method and cause a compiler to generate a proxy object for interacting asynchronously with the client using the callback method, the invention as claimed in claim 1 teaches the specifying of declarative annotations to cause a compiler

to generate persistent components.

In contrast, Ben-Shachar simply teaches a distributed object framework for balancing. While Ben-Shachar does teach a service proxy at the client for making function calls to remote servers and does disclose a callback method associated with the service proxy, both proxy and method belong to the client, not the asynchronous web service being specified. Thus, the service proxy is incapable of reading on the proxy object of claim 1, and the callback method is incapable of reading on the callback method of claim 1.

Even assuming for the sake of argument that the service proxy does read on the proxy object of claim 1, Ben-Shachar simply does not teach a specified member variable which causes a compiler to generate the service proxy. The only compiler disclosed by Ben-Shachar, an IDL compiler, simply generates stub functions for making calls on remote CORBA objects. Nothing is disclosed about such a compiler performing its functions in response to the presence of a member variable. The Examiner even recognizes this on page 4, pointing out that Ben-Shachar does not disclose such a member variable.

Additionally, nothing in Ben-Shachar remotely discloses or suggests specifying declarative annotations to cause a compiler to generate persistent components. Ben-Shachar is focused on the framework in operation and provides very little discussion of its specifying and compilation. In fact, the only such discussion is above-cited discussion of the IDL compiler, which makes no mention of any declarative annotations.

Arulanthu simply does not cure these deficiencies. Arulanthu also discusses an IDL compiler capable of compiling client classes for handling asynchronously CORBA callback. Like Ben-Shachar, the proxy object for receiving callback and the callback method are at the client, not the server, and thus cannot be used by the server for interacting asynchronously with the client. Also, nothing in Arulanthu discloses that a member variable causes the compiler of Arulanthu to generate the proxy object of the

client. Member variables are disclosed as being used for passing parameters, but this is simply not sufficient to teach or suggest “identifying a member variable declared to implement said callback method to cause a compiler to generate a client proxy object for interacting asynchronously with the client,” as is claimed in claim 1.

Further, nothing in Arulanthu teaches or suggests declarative annotations, much less the specifying of such annotations to cause the compiler of Arulanthu to generate one or more persistent components. The only annotations shown in Arulanthu are the comment portions of the various code sections, and none of these annotations are “declarative”, that is, capable of causing the compiler to do something, such as generate persistent components.

Accordingly, claim 1 is patentable over Ben-Shachar and Arulanthu, alone or in combination, under 35 U.S.C. §103(a).

Claims 10, 43, and 52 recite limitations similar to those of claim 1. Accordingly, for at least the same reasons, claims 10, 43, and 52 are patentable over Ben-Shachar and Arulanthu, alone or in combination, under 35 U.S.C. §103(a).

Claims 2 and 44 depend from claims 1 and 43, incorporating their limitations, respectively. Accordingly, for at least the same reasons, claims 2 and 44 are patentable over Ben-Shachar and Arulanthu, alone or in combination, under 35 U.S.C. §103(a).

2. In “Claim Rejections – 35 USC § 103,” item 17 on page 6 of the above-identified Office Action, claims 20-23, 25-29, 62-65, and 67-71 have been rejected as being unpatentable over Arulanthu in view of Ben-Shachar under 35 U.S.C. § 103(a).

Claims 20 and 62 include limitations similar to those of claims 1 and 43, and thus are patentable for at least the same reasons. Accordingly, claims 20 and 62 are patentable over Arulanthu and Ben-Shachar, alone or in combination, under §103(a).

Claims 21-23, 25-29, 63-65, and 67-71 depend from claims 20 and 62, respectively, incorporating their limitations. Consequently, claims 21-23, 25-29, 63-65, and 67-71 are patentable over Arulanthu and Ben-Shachar, alone or in combination, under §103(a).

3. In “Claim Rejections – 35 USC § 103,” item 29 on page 8 of the above-identified Office Action, claims 12-17, 38-42, 54-59, and 80-84 have been rejected as being unpatentable over U.S. Patent No. 6,253,252 to *Schofield* (hereinafter “Schofield”) in view of U.S. Patent Publication No. 2002/0099738 to *Grant* (hereinafter “Grant”) under 35 U.S.C. § 103(a).

*A. Claims 12-17 and 54-59*

Amended claim 12 recites, in “a server having an asynchronous web service, a method comprising:

receiving by the server a message from a client remotely disposed from the

server requesting that the web service be invoked;

parsing by the server the message to identify the requested web service method

in addition to a callback address indicating a location where the client is

listening for callbacks from the web service;

storing by the server the callback address in association with a proxy object; and

invoking by the server the requested web service method, including passing programming language objects as parameters to the web service method, the programming language objects having been mapped from data representation language elements of the message.”

Thus, when amended claim 12 is viewed as a whole, as required by law, it is directed towards a novel method of receiving a client message having data representation language elements, mapping those elements to programming language objects, identifying the method requested by the message and a callback address where the client is listening for callback, storing the callback address in association with a proxy object, and invoking the requested method, including passing the mapped

programming language objects as parameters.

In contrast, Schofield does not teach or suggest a client request having data language elements that are mapped by the server to programming language objects which are then included as parameters of the method requested in the client request. Rather, Schofield is simply directed toward enabling asynchronous responses to distributed object calls between client and server applications communicating through a common execution environment (CEE). The client application may utilize a stub function reflecting the object interface to provide the client-side of the CEE with a service request including a reference to the object, input parameters, and a completion routine address where the client application will accept results of the service request. The CEE may then create a proxy handle structure for the service request, the proxy handle including the object reference and information regarding calls made to the same object. Upon creating the proxy handle, the service request is transferred from the client-side CEE to the server-side CEE. The server-side CEE then selects the appropriate method of the server application to fulfill the service request. That method may fulfill the request itself or call another method to fulfill the request, synchronously or asynchronously. The results of the server application method are then received by the server-side CEE and transferred to the client-side CEE, which may use the completion routine address stored in the proxy handle associated with the request to notify the client application of the results.

The only callback method arguably disclosed by Schofield as part of the web service is the response function. Thus, the only structure in Schofield capable of reading on the proxy object is the proxy handle structure. The proxy handle structure, however, belongs to the client-side CEE. Therefore, the proxy handle structure is incapable of reading on the proxy object in claim 12, because it is the client, not the server which stores the callback address in the proxy handle structure.

But even assuming for the sake of argument that the client-side CEE, server-side CEE, and server, in combination, read on the server of claim 12 (a point that Applicants

do not concede, for the reasons given above), nothing in Schofield teaches that the client request comprise data language elements, or that those elements be mapped by the server to programming language objects which serve as parameters to the requested function.

The mapping of data representation language elements of the message to programming language elements was previously included in claim 18, which was found allowable by the Examiner but for its dependence on claim 12. Since Applicants have now incorporated limitations from claim 18, Applicants respectfully submit that amended claim 12 is in condition for allowance.

Grant simply does not cure the above deficiencies. Rather, Grant discloses client requests to a web server, the web server having a servlet for communicating with a back-end XML engine, including receiving responses from the XML engine. Even assuming for the sake of argument that the servlet receiving responses from the XML engine suggests a proxy object adapted to receive callback from an external web server (an assumption with which the Applicants strongly disagree that need not be addressed at this time), nothing in Grant teaches or suggests that the client request comprise data language elements, or that those elements be mapped by the server to programming language objects which serve as parameters to the requested function.

Accordingly, claim 12 is patentable over Schofield and Grant, alone or in combination, under 35 U.S.C. §103(a).

Claim 54 recites limitations similar to those of claim 12. Accordingly, for at least the same reasons, claim 54 is patentable over Schofield and Grant, alone or in combination, under 35 U.S.C. §103(a).

Claims 13-19 and 55-61 depend from claims 12 and 54, incorporating their limitations, respectively. Accordingly, for at least the same reasons, claims 13-19 and 55-61 are patentable over Schofield and Grant, alone or in combination, under 35

U.S.C. §103(a).

*B. Claims 38-42 and 80-84*

Claim 38 recites, in “a server having a web service, a method comprising:  
receiving by the server a message identifying a callback address, a callback  
method and a proxy object identifier;  
extracting by the server the proxy object identifier from the message;  
determining by the server a method to be invoked based at least in part upon the  
proxy object identifier and the callback method; and  
routing by the server the request to the method to be invoked.”

Thus, when Claim 38 is viewed as a whole, as required by law, it is directed towards a novel method of determining, by a server, a method to invoke based at least in part on a proxy object identifier identified by a message received by the server.

In contrast, Schofield does not teach the determining of an appropriate method based on a proxy object identifier. Even assuming, as is discussed above, that the client-side CEE, combined with the server and server-side CEE, is capable of reading on a “server”, and that the proxy object identifier of claim 38 is read upon by the previously discussed proxy handle, Schofield simply does not teach the determining of the method to invoke based at least in part on the proxy handle. In Schofield, the client-side CEE merely uses the proxy handle as a return identifier to locate the appropriate client. The server and server-side CEE receive and output the handle with the request results, so that the results are returned to the appropriate client, but they do not use the proxy handle to determine which method to invoke. Nothing of the sort is disclosed in Schofield.

Grant fails to cure this deficiency. Even if the servlet is associated with some sort of handle or identifier, that handle or identifier is not used to determine the method to be invoked, as is required by claim 38.



Accordingly, claim 38 is patentable over Schofield and Grant, alone or in combination, under 35 U.S.C. §103(a).

Claim 80 recites limitations similar to those of claim 38. Accordingly, for at least the same reasons, claim 80 is patentable over Schofield and Grant, alone or in combination, under 35 U.S.C. §103(a).

Claims 39-42 and 81-84 depend from claims 38 and 80, incorporating their limitations, respectively. Accordingly, for at least the same reasons, claims 39-42 and 81-84 are patentable over Schofield and Grant, alone or in combination, under 35 U.S.C. §103(a).

4. In “Claim Rejections – 35 USC § 103,” item 44 on page 13 of the above-identified Office Action, claims 30-32, 35-36, 72-74, and 77-78 have been rejected as being unpatentable over Schofield in view of Ben-Shachar under 35 U.S.C. § 103(a).

Claim 30 recites, in “a server offering a web service, a method comprising:  
generating by the server a request to another external web service using a proxy object previously generated by a compiler based upon a service description file associated with the external web service, wherein the request includes a callback address to identify a location to which the external web service should return a response;  
transmitting by the server the request as a request message to the external web service using one or more transmission protocols; and  
receiving by the server an asynchronous response from the external web service.”

Thus, when Claim 30 is viewed as a whole, as required by law, it is directed towards a novel method of generating requests to an external web service by a proxy object of a server having a web service, the proxy object previously generated by a compiler based on a service description file associated with the external web server.

In contrast, Schofield fails to teach or suggest a proxy object previously generated based on a service description file of an external web server. Schofield, as described in part above in item 3, part A of this Response, teaches that a response function of a server application that may be associated with a callback identifier to receive asynchronous responses from an external web server. In embodiments where the server application method calls another asynchronous service to fulfill the request, the server application may include a response function to receive callback from the other asynchronous service, and may associate the response function with a callback identifier to enable the server application to continue operations while waiting for the asynchronous response.

Even if one assumes for the sake of argument that Schofield might suggest some sort of proxy object to listen for asynchronous responses, nothing suggests that such an object need be generated based on a service description file of the external web server. Schofield does not even mention such service description files or contemplate their use as a basis for a proxy object. Further, Schofield explicitly teaches that the server application generates the requests to the external asynchronous web server, thus teaching away from a proxy object making such a request.

Ben-Shachar simply does not cure this deficiency. As discussed above, the server in Ben-Shachar does not even have a proxy object, much less a proxy object generated from a service description file of an external web service.

On page 13 of the Office Action, the Examiner attempts to equate the compiled CORBA IDL, which is linked to the client and server, to the service description file. The IDL, however, only affects client and server communication. As claimed in claim 30, the communication occurs between a server and an external service. Schofield does teach a server and an external server, but no proxy object of the server, much less a proxy objects generated based on a service description file of the external web service. Because the IDL of Schofield does not result in the generation of a proxy object for

communicating with an external web service, the IDL is incapable of reading on the service description file.

Accordingly, claim 30 is patentable over Schofield and Ben-Shachar, alone or in combination, under 35 U.S.C. §103(a).

Claim 72 recites limitations similar to those of claim 30. Accordingly, for at least the same reasons, claim 72 is patentable over Schofield and Ben-Shachar, alone or in combination, under 35 U.S.C. §103(a).

Claims 31-32, 35-36, 73-74, and 77-78 depend from claims 30 and 72, incorporating their limitations, respectively. Accordingly, for at least the same reasons, claims 31-32, 35-36, 73-74, and 77-78 are patentable over Schofield and Ben-Shachar, alone or in combination, under 35 U.S.C. §103(a).

5. In “Claim Rejections – 35 USC § 103,” item 53 on page 15 of the above-identified Office Action, claims 33-34, 37, 75-76, and 79 have been rejected as being unpatentable over Schofield in view of Ben-Shachar and further in view of Grant under 35 U.S.C. § 103(a).

Grant does not cure the deficiencies of Schofield and Ben-Shachar. Thus, claims 30 and 72 remain patentable even when Schofield, Ben-Shachar, and Grant are combined.

Claims 33-34, 37, 75-76, and 79 depend from claims 30 and 72, respectively, incorporating their limitations. Consequently, claims 33-34, 37, 75-76, and 79 are patentable over Schofield, Ben-Shachar, and Grant, alone or in combination, under §103(a).

Allowable Subject Matter

Applicants thank the Examiner for finding claims 4-9, 18-19, 46-51, and 60-61 allowable but for their dependence on rejected base claims. For the reasons given above, Applicants now believe that both these claims and their rejected base claims are allowable.

Conclusion

In view of the foregoing, reconsideration and allowance of claims 1-2, 4-10, 12-23, 25-44, 46-52, 54-65, and 67-84 are solicited. Applicant submits that claims 1-2, 4-10, 12-23, 25-44, 46-52, 54-65, and 67-84 are in condition for allowance. Accordingly, a Notice of Allowance is respectfully requested. If the Examiner has any questions concerning the present paper, the Examiner is kindly requested to contact the undersigned at (206) 407-1513. If any fees are due in connection with filing this paper, the Commissioner is authorized to charge the Deposit Account of Schwabe, Williamson and Wyatt, P.C., No. 50-0393.

Respectfully submitted,  
SCHWABE, WILLIAMSON & WYATT, P.C.

Dated: February 7, 2007

/Robert C. Peck/  
Robert C. Peck  
Reg. No. 56,826

Schwabe, Williamson & Wyatt, P.C.  
Pacwest Center, Suites 1600-1900  
1211 SW Fifth Avenue  
Portland, Oregon 97204  
Telephone: 503-222-9981